

# BD2 - PROJEKT

## Przetwarzanie własnych agregatów CLR UDA

Joanna Hełdak

13.06.2025r.

### Spis treści

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Opis problemu</b>                                     | <b>1</b> |
| <b>2</b> | <b>Udostępnione funkcjonalności</b>                      | <b>2</b> |
| 2.1      | Pokaż tabelkę . . . . .                                  | 2        |
| 2.2      | Pokaż statystyki globalne . . . . .                      | 2        |
| 2.3      | Porównaj statystyki dla regionów . . . . .               | 3        |
| 2.4      | Przeanalizuj statystyki ręcznie . . . . .                | 3        |
| <b>3</b> | <b>Typy danych oraz metody udostępnione w ramach API</b> | <b>4</b> |
| 3.1      | 5 agregatów . . . . .                                    | 4        |
| 3.2      | Metody wykorzystywane w API . . . . .                    | 4        |
| <b>4</b> | <b>Implementacja</b>                                     | <b>5</b> |
| 4.1      | Struktura . . . . .                                      | 5        |
| 4.2      | API złożone z dwóch komponentów . . . . .                | 5        |
| <b>5</b> | <b>Prezentacja przeprowadzonych testów jednostkowych</b> | <b>6</b> |
| <b>6</b> | <b>Podsumowanie i wnioski</b>                            | <b>6</b> |
| <b>7</b> | <b>Kod źródłowy - Link</b>                               | <b>6</b> |
| <b>8</b> | <b>Literatura</b>  | <b>7</b> |

### 1 Opis problemu

Celem projektu było zaprojektowanie i zaimplementowanie API wspierającego zestaw pięciu agregatów zdefiniowanych przez użytkownika UDA przy użyciu technologii CLR. Trzy z tych agregatów zostały przygotowane w wersjach rozszerzonych. Agregaty są hostowane w SQL Server i ich przetwarzanie odbywa się po stronie serwera. API udostępnia metody do obsługi agregatów z poziomu aplikacji *C#*, hermetyzując szczegóły zapytań SQL. Całość jest dostępna z aplikacji konsolowej, która przedstawia przykładowe zastosowanie agregatów. Zarówno logika API, jak i agregaty CLR zostały przetestowane z użyciem testów jednostkowych.

## 2 Udostępnione funkcjonalności

```
----- Menu Główne -----
1. Pokaż tabelkę
2. Pokaż statystyki globalne
3. Porównaj statystyki dla regionów
4. Przeanalizuj statystyki ręcznie
q. Wyjście z programu
_
```

Rysunek 1: Menu Główne aplikacji konsolowej

Użytkownik na początku uzyskuje dostęp to przejrzania tabeli wskaźnika szczęścia, zobaczenia gotowych statystyk, zobaczenia statystyk porównawczych bądź możliwość do własnej analizy statystyk. Opcje pokolei:

### 2.1 Pokaż tabelkę

Użytkownik po wyborze obszaru, dla którego dane chciałby zobaczyć, dostaje wynik w postaci tabeli:

```
Wybierz region:
1. Central and Eastern Europe
2. Commonwealth of Independent States
3. East Asia
4. Latin America and Caribbean
5. Middle East and North Africa
6. North America and ANZ
7. South Asia
8. Southeast Asia
9. Sub-Saharan Africa
10. Western Europe
11. Wszystkie regiony
(r - powrót, q - wyjście z programu)
7
-----
| ID | Kraj      | Region      | Wskaźnik Szczęścia |
-----
| 87 | Nepal     | South Asia  | 5,269               |
-----
| 89 | Maldives  | South Asia  | 5,198               |
-----
| 101 | Bangladesh | South Asia  | 5,025               |
-----
| 105 | Pakistan  | South Asia  | 4,934               |
-----
| 129 | Sri Lanka  | South Asia  | 4,325               |
-----
| 139 | India     | South Asia  | 3,819               |
-----
| 149 | Afghanistan | South Asia  | 2,523               |
-----
Count: 7
Naciśnij dowolny klawisz, aby powrócić do menu...
_
```

Rysunek 2: Opcja 1 - wybieramy z jakiego zakresu chcemy zobaczyć dane

### 2.2 Pokaż statystyki globalne

Użytkownik ma możliwość wyboru, jaki rodzaj gotowych statystyk chce wybrać i dostaje wynik w formie tabeli:

```

1. Ogólne
2. Pogrupowane według regionu
(r - powrót, q - wyjście z programu)
2

```

| Region                             | Ilość krajów | Mediana | Moda | Std   | Przedział | Kwantyl 0.25 | Kwantyl 0.75 | Min   | Max   |
|------------------------------------|--------------|---------|------|-------|-----------|--------------|--------------|-------|-------|
| Central and Eastern Europe         | 17           | 6,078   | 5,1  | 0,493 | 1,864     | 5,813        | 6,255        | 5,101 | 6,965 |
| Commonwealth of Independent States | 12           | 5,5055  | 6,18 | 0,438 | 1,304     | 5,171        | 5,744        | 4,875 | 6,179 |
| East Asia                          | 6            | 5,8925  | 5,68 | 0,44  | 1,245     | 5,477        | 5,94         | 5,339 | 6,584 |
| Latin America and Caribbean        | 20           | 6,0365  | 4,89 | 0,693 | 3,454     | 5,764        | 6,309        | 3,615 | 7,069 |
| Middle East and North Africa       | 17           | 4,887   | 4,92 | 0,999 | 3,499     | 4,584        | 6,106        | 3,658 | 7,157 |
| North America and ANZ              | 4            | 7,23    | 7,18 | 0,138 | 0,326     | 7,103        | 7,183        | 6,951 | 7,277 |
| South Asia                         | 7            | 4,934   | 2,52 | 0,993 | 2,746     | 4,325        | 5,025        | 2,523 | 5,269 |
| Southeast Asia                     | 9            | 5,384   | 6,38 | 0,606 | 1,951     | 5,03         | 5,88         | 4,426 | 6,377 |
| Sub-Saharan Africa                 | 36           | 4,6305  | 3,62 | 0,655 | 2,904     | 4,107        | 4,984        | 3,145 | 6,049 |
| Western Europe                     | 21           | 7,085   | 7,32 | 0,657 | 2,306     | 6,491        | 7,392        | 5,536 | 7,842 |

```

Count: 10
Naciśnij dowolny klawisz, aby powrócić do menu...

```

Rysunek 3: Opcja 2

## 2.3 Porównaj statystyki dla regionów

Użytkownik w tej sekcji, najpierw podaje ile regionów chce porównać, a następnie wybiera regiony do porównania. Jako wynik dostaje tabelę porównawczą, podobną jak w podpunkcie 2.2 (Rysunek 3).

```

Liczba regionów do porównania:
(r - powrót, q - wyjście z programu)
3_

```

Rysunek 4: Opcja 3 - wpisujemy ile regionów do porównanie

```

Dostępne regiony:
1. Central and Eastern Europe
2. Commonwealth of Independent States -- WYBRANY
3. East Asia
4. Latin America and Caribbean
5. Middle East and North Africa
6. North America and ANZ
7. South Asia
8. Southeast Asia
9. Sub-Saharan Africa
10. Western Europe -- WYBRANY
Wybierz region (3 z 3):
_

```

Rysunek 5: Opcja 3 - wybieramy regiony do porównania

## 2.4 Przeanalizuj statystyki ręcznie

Tutaj użytkownik ma możliwość ręcznego ustawienia, najpierw regionu dla które chce dostać statystyki (jak na Rysunku 5), potem wybiera rodzaj agregatu UDA, w przypadku Kwantyla, musi podać również rząd kwantyla. Jako odpowiedź dostaje wartość double z wyliczonym wynikiem.

```

Wybrany region: Western Europe
Wybierz agregat:
1. Mode
2. StdDev
3. Median
4. Quantile
5. Range
(r - powrót, q - wyjście z programu)
4
Podaj wartość kwantyla [0 - 1]:
0.2
Wynik dla Western Europe --- Quantile : 6,483
Naciśnij dowolny klawisz, aby powrócić do menu...

```

Rysunek 6: Opcja 4

### 3 Typy danych oraz metody udostępnione w ramach API

#### 3.1 5 agregatów

1. **Odchylenie Standardowe(StdDev)**  
Agregat obliczający odchylenie standardowe, będące miarą rozproszenia wartości wokół średniej.
2. **Mediana (Median) - w wersji rozszerzonej**  
Agregat wyznaczający medianę wartości numerycznych, sortując dane i wybierając środkowy element lub średnią z dwóch środkowych.
3. **Rozstęp (Range)**  
Agregat zwracający zakres (różnicę między maksymalną a minimalną wartością) w zbiorze danych liczbowych.
4. **Kwantyl (Quantile) - w wersji rozszerzonej**  
Agregat wyliczający wskazany kwantyl (np. kwartyl) z danych liczbowych, umożliwiając dynamiczne określenie poziomu kwantyla przez użytkownika.
5. **Moda (Mode) - w wersji rozszerzonej**  
Agregat obliczający modę, czyli wartość najczęściej występującą w zestawie danych, z uwzględnieniem zaokrąglenia do 3 liczb po przecinku.

#### 3.2 Metody wykorzystywane w API

1. GetAllRegions()  
Zwraca listę wszystkich dostępnych regionów znajdujących się w tabeli HappinessTab, wykorzystując procedurę składowaną *uspGetRegions*
2. GetAllAggregatesNames()  
Zwraca listę wszystkich dostępnych UDA CLR znajdujących się w tabeli HappinessTab, wykorzystując procedurę składowaną *uspGetAggregatesNames*
3. getTabData(string regionName = null)  
Zwraca rekordy tabeli HappinessTab z bazy. Umożliwia opcjonalne filtrowanie po regionie.
4. getStatistics(string name = null)  
Generuje zestawienie statystyczne na podstawie wyników wszystkich agregatów UDA dla całego zbioru danych lub wybranego regionu.
5. getStatisticsGrouped()  
Generuje zestawienie statystyczne na podstawie wyników wszystkich agregatów UDA dla całego zbioru danych pogrupowanego według regionu.
6. GetCustomAggregateForRegion(string selectedAggregate, string selectedRegion, double? quantile = null)  
Oblicza wybraną przez użytkownika funkcję agregującą dla wskazanego przez użytkownika regionu, z możliwością podania konkretnego poziomu kwantyla. (Domyślnie kwantyl ustawiony jest na 0.5)

## 4 Implementacja

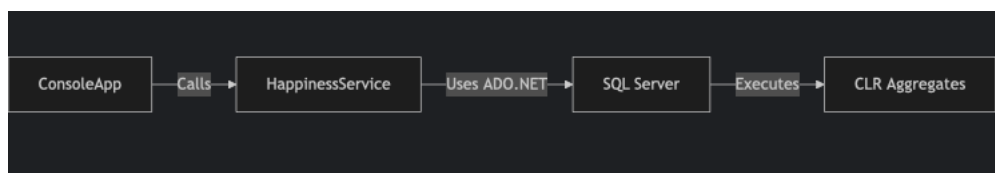
### 4.1 Struktura

Struktura Solution składa się z 4 projektów:

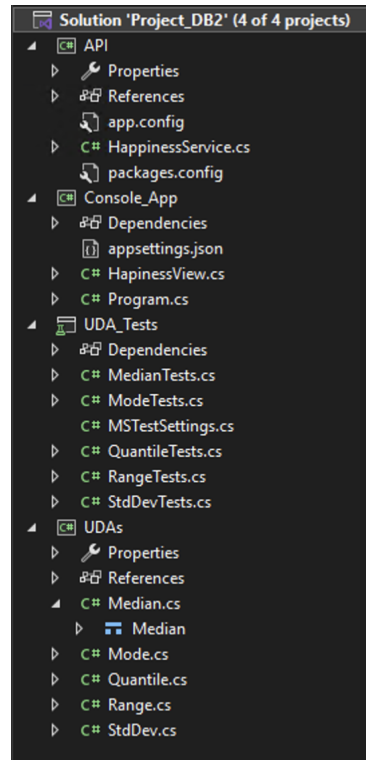
1. UDAs  
Projekt typu Class Library zawiera zdefiniowane przeze mnie agregaty CLR (User-Defined Aggregates). Po skompilowaniu projektu do formatu .dll, utworzony assembly został zaimportowany do środowiska Microsoft SQL Server.
2. UDA Tests  
Projekt przeznaczony do testowania napisanych UDA. Dla każdego zdefiniowanego agregatu przygotowałam osobny zestaw testów jednostkowych. Pozwoliły mi ona na weryfikację poprawności działania każdej metody oraz wychwycenie ewentualnych błędów we wczesnym etapie prac.
3. API  
Projekt typu Class Library zawierający potrzebne metody do komunikacji z bazą SQL Server.
4. Console App  
Projekt typu Console Application służy jako warstwa aplikacyjna umożliwiająca komunikację z bazą danych przy użyciu technologii ADO.NET. Odpowiada za logikę aplikacji oraz interakcję z użytkownikiem. Program wywołuje komunikuje się z warstwą API, która z kolei korzysta z procedur składowanych oraz funkcji agregujących zdefiniowanych po stronie bazy danych.

### 4.2 API złożone z dwóch komponentów

1. **API na poziomie bazy danych - UDAs**  
Są to agregaty CLR napisane w *C#*, działające wewnątrz SQL Servera, wywoływane bezpośrednio z T-SQL
2. **API na poziomie aplikacji - HappinessService.cs**  
Klasa *C#* która wykorzystuje UDAs przez ADO.NET. Zawiera logikę biznesową.

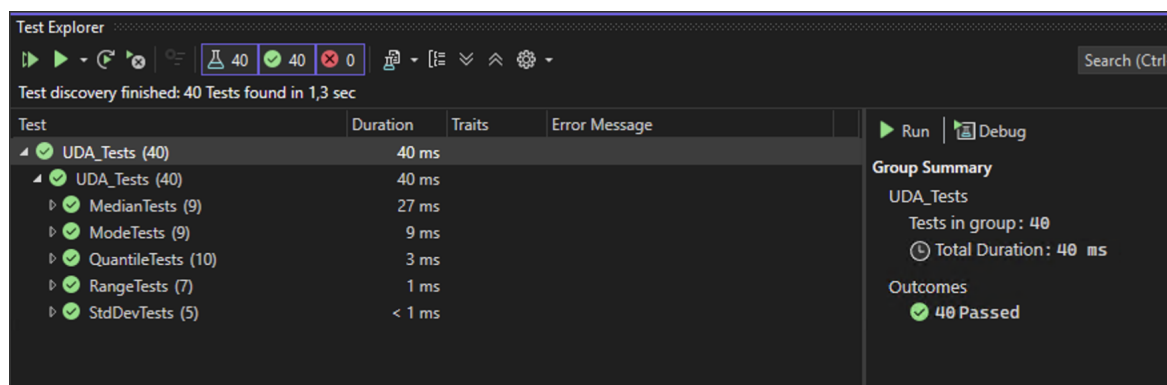


Rysunek 7: Diagram Funkcjonowania API



Rysunek 8: Solution

## 5 Prezentacja przeprowadzonych testów jednostkowych



Rysunek 9: Wyniki Testów Jednostkowych

## 6 Podsumowanie i wnioski

Udało się stworzyć kompletny zestaw pięciu agregatów, w tym przynajmniej dwa w wersji rozszerzonej. Agregaty zostały skutecznie zaimplementowane, przetestowane i zintegrowane z bazą danych. Dzięki podziałowi na osobne projekty udało się zachować modularność kodu oraz umożliwić jego łatwą rozbudowę i utrzymanie.

## 7 Kod źródłowy - Link

Kod źródłowy oprócz doślanego zip-a, jest również dostępny na GitHub pod adresem:

## 8 Literatura

- (a) instruktaż na YT dotyczący unit testów w *C#*
- (b) AGH laboratoria instrukcje
- (c) Microsoft Docs: CLR UDAs